INDUSTRIAL TRAINING REPORT

CodeAlpha

Submitted in partial fulfillment of the Requirements
for the award of

Degree of Bachelor of Technology in
Computer Science & Engineering

Submitted By : ADITI SINGH

University Roll No : 23210101548

Semester/Branch : 3/CSE

Submitted to : RIYA KUKRETI
STUTI BHATT

Department of Computer Science & Engineering
UIT, UTTARANCHAL UNIVERSITY Dehradun
(Uttarakhand), 248001

CERTIFICATE OF COMPLETION

DECLARATION

I hereby declare that the Industrial Training Report on C++ PROGRAMMING, Industry is an authentic record of my own work as requirements of Minor/ Major Industrial Training during the period from 1st October 2024 to 31st October 2024 for the award of degree of B.Tech. (Computer Science & Engineering), UIT, Uttaranchal University, Dehradun (U.K.), under the guidance of Ms. Stuti Bhatt and Ms. Riya Kukreti.

Date: _____

ADITI SINGH
23210101548

Certified that the above statement made by the student is correct to the best of our knowledge and belief

Examined by:

Ms. Stuti Bhatt

Ms. Riya Kukreti

Head of Department

ACKNOWLEDGEMENT

First and foremost, I wish to express my sincere thanks and gratitude to my esteemed Mentor “CodeAlpha” who has contributed so much for successful completion of my Industrial Training by his thoughtful reviews and valuable guidance.

Next, I would like to tender my sincere thanks to “Prof. Madhu Kirola” (Head of Computer Science & Engineering Department) for his co-operation and encouragement.

ADITI SINGH
23210101548

OVERVIEW OF CodeAlpha

CodeAlpha offers online programs in C++ programming, web development, app development, Java programming, and Python programming. These programs include real projects, learning by doing, and flexibility to learn at your own pace. The platform is designed to allow learners to gain practical experience at their own pace, with a strong focus on remote, online learning. CodeAlpha boasts a large community of learners and successful internships, with over 1,29,000 students having completed internships through their platform.

TABLE OF CONTENTS

Chapter 1:
Introduction (An overview of whole report)

This report presents an overview of my C++ programming-focused industrial training and internship, where I applied coding skills to develop real-world applications. The goal of this training was to gain hands-on experience in software development using C++, a powerful language that's widely used in industries like gaming, system software, and application development.

1.1 Internship Overview

This report presents an overview of my industrial training and internship focused on practical applications in C++ programming.

During this internship, I applied my skills to develop three real-world projects, gaining essential experience in development. The primary goal was to strengthen software my understanding of C++ and its applications in real-world scenarios, from system software to gaming. **Key Projects in C++:**

- **CGPA Calculator:** This application calculates students' CGPA based on their grades and credit hours, helping me learn data processing and user input handling.
- **Sudoku Solver:** A puzzle-solving program that uses recursion and backtracking, improving my algorithm skills and logical thinking.
- **Login and Registration Form:** A user login system with secure credential handling, which enhanced my skills in basic data security and file management.

Each project presented unique challenges, allowing me to practice problem-solving, coding standards, and debugging methods. Overall, this training expanded my skills in C++ and prepared me for future projects in tech development

Chapter 2: Industrial Training

2.1 Objectives The goal of my industrial training was to build a strong foundation in C++ programming, focusing on hands-on experience to develop practical and industry-relevant skills. This report highlights the key objectives and skills gained during this training period.

Strengthen Technical Skills in C++ The training aimed to deepen my technical expertise in C++, helping me master programming concepts and write efficient code. Through various projects, I applied my knowledge to create reliable applications, reinforcing my understanding of C++ fundamentals.

Enhance Problem-Solving Abilities Problem-solving was a key focus, as each project presented unique challenges. This experience sharpened my logical thinking, as I learned to analyze requirements, troubleshoot issues, and devise efficient solutions.

Understand Component Integration I also focused on understanding how different parts of an application interact. This gave me insights into system integration, which is crucial for building cohesive software.

Professional Project Workflow Learning to manage project timelines, collaborate effectively, and follow structured workflows added real-world value to my training, preparing me for professional development environments.

2.2 Tools and Technology Used

During my internship, I used several essential tools and technologies that supported my development process and improved my efficiency.

- **Programming Language (C++):** C++ was the core language for all my projects. It enabled me to apply fundamental programming concepts, manage complex data structures, and develop logic-driven applications. This versatility was crucial for creating projects like the CGPA calculator, Sudoku solver, and login system, allowing me to deepen my understanding of object-oriented programming principles.

- **Development Environment (Visual Studio Code):** Visual Studio Code (VS Code) served as my primary Integrated Development Environment (IDE). Its user-friendly interface, syntax highlighting, and debugging tools made coding more efficient. The availability of extensions further enhanced productivity, particularly for larger projects, enabling me to navigate and manage code effectively.

- **Version Control (GitHub):** GitHub was vital for version control, helping me manage project versions and track changes securely. This platform allowed me to organize my work, ensuring easy access to different project stages. Additionally, GitHub provided experience with industry best practices in code management and collaboration

2.3 Techniques Used in Different Departments

Throughout my internship, I applied key techniques across different projects—the CGPA calculator, Sudoku solver, and login and registration form—to enhance quality and efficiency.

- **Modular Coding:** For the CGPA calculator, I used modular coding to break down tasks into functions, improving readability and optimizing performance for complex calculations.
- **Systematic Testing:** In the Sudoku solver, I conducted rigorous testing and debugging, creating test cases to ensure accurate solutions and a smooth user experience.
- **User-Centered Design:** For the login and registration form, I applied user-centered design techniques to make navigation intuitive and improve usability.
- **Data Security:** In the login system, I implemented secure data handling, including encryption and authentication, to protect user information and ensure data integrity.

2.4 Software and Tools Used

During my internship, I utilized several key software tools that supported efficient coding, organization, and project management.

- **Visual Studio Code (VS Code):** VS Code was my primary environment for writing, testing, and debugging C++ code. Its user-friendly interface, syntax highlighting, and debugging capabilities helped me improve code quality while efficiently managing project files.

- **GitHub for Version Control:** GitHub was essential for version control, allowing me to manage code versions, track changes, and keep my work organized.This tool helped me understand the importance of structured coding and file management, crucial skills for collaborative development environments.

- **GCC Compiler:** The GCC compiler enabled efficient code compilation and error-checking, making it invaluable in refining and optimizing my C++ code for projects like the CGPA calculator and Sudoku solver

.

- **Notion for Task Management:** I used Notion to organize tasks and set project milestones, which kept me on track with deadlines and provided a clear view of each project’s requirements

2.5 Highlights of Training Exposure

During my C++ internship, various software tools and technologies were utilized to streamline development, ensure code quality, and enhance collaboration. I worked on exciting projects like a CGPA calculator, Sudoku solver, and a login form, which helped me apply these tools effectively in real-world applications

- **Foundation in C++ Programming:**
  During the internship, I gained hands-on experience in C++ programming through projects like a CGPA calculator, a Sudoku solver, and a login and registration form. Each project helped me improve my understanding of C++ fundamentals, coding standards, and software development practices.

  Coding Standards and Modularity:
- I learned the importance of writing clean, organized code and following coding standards, which made my code easier to read, debug, and modify. Using modular coding, I divided complex tasks into smaller functions, especially beneficial in larger projects like the CGPA calculator.

- **Systematic Debugging Skills:**
  Debugging was essential to project quality, especially for the Sudoku solver. I used step-by-step debugging and error analysis to identify and resolve issues, which improved my problem-solving abilities and taught me effective debugging techniques for future projects.

  Focus on Data Handling and Security:
- In the login and registration form, I implemented data handling and basic security practices, such as encryption and authentication, to protect user data. This introduced me to the importance of data security and privacy, crucial in developing secure applications.

- **Project Management and Organization:**

  Using project management tools like Notion, I learned how to organize tasks, set goals, and track milestones. This approach helped me stay organized, manage time effectively, and keep up with project deadlines.

- **Version Control with GitHub:**

  I used GitHub for version control, allowing me to track changes, revert to previous versions if needed, and manage my code efficiently. This experience taught me the importance of version control and collaboration, essential in professional development environments.

- **User-Centered Design:**

  Creating user-friendly interfaces was a key focus, especially for the login form. I applied user-centered design principles to make navigation simple and intuitive, which enhanced the usability of the application.

- **Logical Thinking and Problem-Solving:**

  Developing the Sudoku solver required strong logical thinking and efficient algorithms. Tackling this project sharpened my problem-solving skills, which I can now apply to other complex programming challenges.

- **Practical Knowledge for Industry Preparation:**

  This internship gave me valuable insights into industry practices, coding standards, and professional tools, preparing me for future software development roles. Working on real-world applications strengthened my skills and boosted my confidence in applying C++ to solve practical problems.

Chapter3: Problem Identification/Case Study:

During my C++ internship, I encountered several challenges while working on three key projects: a CGPA calculator, a Sudoku solver, and a login and registration form. Each project presented unique problems that required careful analysis and innovative solutions.

Project 1: CGPA Calculator

Challenge:

- The primary challenge in developing the CGPA calculator was ensuring accurate input validation.
- Users needed to enter valid grades and credit hours, and any incorrect data could lead to faulty calculations.

Solution: To tackle this, I implemented input validation techniques. I created functions that checked whether the input was numeric and within an acceptable range. Additionally, I provided clear error messages to guide users in correcting their input. This not only improved the calculator's reliability but also enhanced the overall user experience.

Outcome:

- As a result, the CGPA calculator accurately computed grades and provided feedback, helping users understand how their inputs affected their overall performance.

Project 2: Sudoku Solver

Challenge:

- The Sudoku solver posed challenges in efficiently solving complex puzzles.
- The backtracking algorithm was the foundation of the solution, but it could be slow for more difficult grids.

o**S**pt**o**i**l**m**ut**iz**io**a**n**ti**:**o n I tecrhesneiqaurcehse dt o aendh anicmep ltehmee npterdf ormance of the backtracking algorithm. By incorporating strategies like
eliminating impossible values and prioritizing empty cells, I significantly reduced the solving time. Additionally, I designed an intuitive user interface to allow users to easily input their puzzles and view solutions.

Outcome:

- The optimized Sudoku solver quickly and accurately solved puzzles, providing users with a seamless experience. User
- feedback indicated that the interface was clear and easy to navigate, which added to the enjoyment of solving Sudoku.

Project 3: Login and Registration Form

Challenge:

- Developing a secure login and registration form was essential to protect user data.
- The challenge was to implement secure data storage and ensure users created strong passwords

Solution: I researched encryption methods and decided to use hashing techniques to store passwords securely. I also implemented input validation to enforce strong password requirements, such as a minimum length and a mix of characters. This process involved providing clear guidelines to users, ensuring they understood how to create secure passwords.

Outcome:

- The login and registration form was secure and user-friendly.
- Users could create accounts with confidence, knowing their information was protected.
- The validation messages guided them in creating strong passwords, improving overall security.

Chapter 4: Recommendations

Based on my internship experience, here are some recommendations that could make future programs more effective and enjoyable for interns:

- **Provide Advanced C++ Training:** Offering additional training in advanced C++ topics, such as data structures, pointers, and algorithm optimization, would help interns gain deeper technical skills and prepare them for more complex tasks.
- **Expand Project Scope:** Extending projects to include additional features or levels of difficulty can make them more engaging. For example, adding secure authentication to the login system or handling more challenging Sudoku puzzles could help interns tackle real-world problems with increased complexity.
- **Regular Code Reviews:** Introducing regular code reviews with mentors or peers would allow interns to receive feedback, learn coding best practices, and improve their problem-solving abilities.
- **Encourage Continuous Practice:** Regular hands-on practice with various C++ projects could help interns strengthen their skills and build adaptability, especially when facing new or unique coding challenges.
- **Teach Debugging Techniques:** Teaching interns the basics of debugging tools and testing methods can help them identify and fix errors quickly, improving the quality of their code.

Conclusion Overall, my C++ internship provided valuable learning opportunities, enhancing both my technical skills and understanding of project-based development. Implementing these recommendations could make future programs even more effective by helping interns gain confidence and build skills essential for software development. These enhancements would better prepare interns for professional roles, equipping them with both practical experience and a solid technical foundation

References

CGPA Calculation in C++ https://www.geeksforgeeks.org/cpp-program-to-calculate-cgpa/

Sudoku Solver Algorithm

https://www.geeksforgeeks.org/sudoku-backtracking-7/

Implementing Sudoku Solver in C++
https://www.tutorialspoint.com/sudoku-solver-backtracking

Login and Registration System in C++ https://www.myprojectideas.com/cpp-projects/login-and-registration-system-in-cpp/

Appendices Snapshots of the project

Project :- 1 CGPA Calculator

```cpp
// CGPA  CALCULATOR
#include <iostream>
using namespace std;

int main() {
    // Variables to store credit hours and grade points for each subject
    float creditHours[] = {3, 4, 3, 2}; // Credit hours for the subjects
    float gradePoints[] = {8.5, 9.0, 9.2, 7.8}; // Grade points for the subjects

    // Variables to hold total grade points and total credit hours
    float totalGradePoints = 0, totalCredits = 0, cgpa;

    int numSubjects = sizeof(creditHours) / sizeof(creditHours[0]); // Number of subjects

    // Loop through each subject to calculate total grade points and total credits
    for (int i = 0; i < numSubjects; i++) {
        totalGradePoints += gradePoints[i] * creditHours[i]; // Add weighted grade points
        totalCredits += creditHours[i]; // Add credit hours
    }

    // CGPA calculation
    if (totalCredits != 0) {
        cgpa = totalGradePoints / totalCredits;
        cout << "Your CGPA is: " << cgpa << endl;
    } else {
        cout << "Error: Total credits cannot be zero." << endl;
    }

    return 0;
}
```

Output :-

```
PS C:\Users\aditi> cd "c:\Users\aditi\.vscode\taks1.cpp\" ; if ($?) { g++ tasks1.cpp -o tasks1 } ; if ($?) { .\tasks1 }
Your CGPA is: 8.725
PS C:\Users\aditi\.vscode\taks1.cpp>
```

Project :- 2 Sudoku Solver

tasks2.cpp > ...

```cpp
// SUDUKO SOLVER
#include <iostream>
using namespace std;

#define N 9 // Size of the Sudoku grid

// Function to print the Sudoku grid
void printGrid(int grid[N][N]) {
    for (int row = 0; row < N; row++) {
        for (int col = 0; col < N; col++)
            cout << grid[row][col] << " ";
        cout << endl;
    }
}

// Check if placing a number in a cell is safe (row, column, and 3x3 grid)
bool isSafe(int grid[N][N], int row, int col, int num) {
    // Check if the number is in the current row
    for (int x = 0; x < N; x++)
        if (grid[row][x] == num)
            return false;

    // Check if the number is in the current column
    for (int x = 0; x < N; x++)
        if (grid[x][col] == num)
            return false;

    // Check the 3x3 sub-grid
    int startRow = row - row % 3, startCol = col - col % 3;
    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++)
            if (grid[i + startRow][j + startCol] == num)
                return false;

    return true;
}
```

```cpp
}

// Function to solve the Sudoku using backtracking
bool solveSudoku(int grid[N][N], int row, int col) {
    // If we've reached the last cell, the puzzle is solved
    if (row == N - 1 && col == N)
        return true;

    // Move to the next row if we've reached the end of the current row
    if (col == N) {
        row++;
        col = 0;
    }

    // Skip the cells that are already filled
    if (grid[row][col] != 0)
        return solveSudoku(grid, row, col + 1);

    // Try placing numbers 1 to 9 in the current cell
    for (int num = 1; num <= 9; num++) {
        if (isSafe(grid, row, col, num)) {
            grid[row][col] = num; // Place the number

            // Recursively solve the rest of the grid
            if (solveSudoku(grid, row, col + 1))
                return true;
        }

        // Undo the assignment (backtrack)
        grid[row][col] = 0;
    }

    // Trigger backtracking
    return false;
}
```

```
    return false;
}

int main() {
    // Initial Su
    int grid[N][N
        {3, 0, 6, 5, 0, 8, 4, 0, 0},
        {5, 2, 0, 0, 0, 0, 0, 0, 0},
        {0, 8, 7, 0, 0, 0, 0, 3, 1},
        {0, 0, 3, 0, 1, 0, 0, 8, 0},
        {9, 0, 0, 8, 6, 3, 0, 0, 5},
        {0, 5, 0, 0, 9, 0, 6, 0, 0},
        {1, 3, 0, 0, 0, 0, 2, 5, 0},
        {0, 0, 0, 0, 0, 0, 0, 7, 4},
        {0, 0, 5, 2, 0, 6, 3, 0, 0}
    };

    // Solve the Sudoku puzzle and print the result
    if (solveSudoku(grid, 0, 0)) {
        cout << "Solved Sudoku Grid:\n";
        printGrid(grid);
    } else {
        cout << "No solution exists\n";
    }

    return 0;
}
```

int [9]{(int)3, (int)0, (int)6, (int)5, (int)0, (int)8, (int)4, (int)0, (int)0}

Output :-

```
PS C:\Users\aditi\.vscode\taks1.cpp> cd "c:\Users\aditi\.vscode\taks1.cpp\" ; if ($?) { g++ tasks2.cpp -o tasks2 } ; if ($?) { .\tasks2 }
Solved Sudoku Grid:
3 1 6 5 7 8 4 9 2
5 2 9 1 3 4 7 6 8
4 8 7 6 2 9 5 3 1
2 6 3 4 1 5 9 8 7
9 7 4 8 6 3 1 2 5
8 5 1 7 9 2 6 4 3
1 3 8 9 4 7 2 5 6
6 9 2 3 5 1 8 7 4
7 4 5 2 8 6 3 1 9
PS C:\Users\aditi\.vscode\taks1.cpp>
```

Pro ject :- 3 Login and Registration Form

```cpp
                                        //LOGIN AND REGISTRATION FORM
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

// Function to register a new user
void registerUser() {
    string username, password;
    cout << "Enter a username: ";
    cin >> username;
    cout << "Enter a password: ";
    cin >> password;

    // Open the file in append mode to store user credentials
    ofstream file;
    file.open("credentials.txt", ios::app);
    if (file.is_open()) {
        file << username << " " << password << endl;
        file.close();
        cout << "Registration successful!" << endl;
    } else {
        cout << "Error opening file!" << endl;
    }
}

// Function to login an existing user
bool loginUser() {
    string username, password, storedUsername, storedPassword;
    cout << "Enter your username: ";
    cin >> username;
    cout << "Enter your password: ";
    cin >> password;

    // Open the file in read mode to verify user credentials
    ifstream file;
```

```cpp
    ifstream file;
    file.open("credentials.txt");
    if (file.is_open()) {
        bool loginSuccess = false;
        while (file >> storedUsername >> storedPassword) {
            if (storedUsername == username && storedPassword == password) {
                loginSuccess = true;
                break;
            }
        }
        file.close();

        if (loginSuccess) {
            cout << "Login successful!" << endl;
            return true;
        } else {
            cout << "Invalid username or password!" << endl;
            return false;
        }
    } else {
        cout << "Error opening file!" << endl;
        return false;
    }
}

int main() {
    int choice;

    while (true) {
        cout << "1. Register\n2. Login\n3. Exit\nChoose an option: ";
        cin >> choice;

        if (choice == 1) {
            registerUser();
        } else if (choice == 2) {
```

```
        if (choice == 1) {
            registerUser();
        } else if (choice == 2) {
            if (loginUser()) {
                break;  // Exit loop if login is successful
            }
        } else if (choice == 3) {
            cout << "Exiting..." << endl;
            break;
        } else {
            cout << "Invalid option! Try again." << endl;
        }
    }

    return 0;
}
```

Output :-

CONCLUSION :-

In conclusion, my internship in C++ programming allowed me to explore and enhance my technical skills by working on practical projects like the CGPA Calculator, Sudoku Solver, and Login & Registration Form. Each project presented unique challenges, from implementing input validation and algorithm optimization to ensuring secure data handling. These experiences helped me gain a deeper understanding of coding principles, problem-solving, and user-centered design.

The internship also introduced me to essential tools and techniques that improved my programming efficiency and project execution. It reinforced the importance of testing, debugging, and delivering solutions that are both functional and user-friendly. This learning journey has strengthened my foundation in software development and prepared me for tackling more advanced projects in the future. Overall, it was a valuable experience that bridged theoretical knowledge with real-world application, boosting my confidence as a developer.